A discrete 3D+t Laplacian framework for mesh animation processing

Franck Hétroy

A discrete 3D+t Laplacian framework for mesh animation processing

Franck Hétroy

ISSN 0249-6399 ISRN INRIA/RR--8003--FR+ENG

RESEARCH REPORT
N° 8003
June 2012
Project-Team Morpheo

A discrete 3D+t Laplacian framework for mesh animation processing

Franck Hétroy*

Project-Team Morpheo

Research Report n° 8003 — June 2012 — 17 pages

Abstract: In this report we extend the discrete 3D Laplacian framework to mesh animations, represented as temporally coherent sequences of meshes. In order to let the user control the motion influence with respect to the geometry, we introduce a parameter for the time dimension. Our discrete 3D+t Laplace operator holds the same properties as the discrete 3D Laplacian, as soon as this parameter is non negative. We demonstrate the usefulness of this framework by extending Laplacian-based mesh editing and fairing techniques to mesh animations.

Key-words: mesh animation, Laplacian, mesh editing, mesh fairing

* Laboratoire Jean Kuntzmann, Inria, Université de Grenoble

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Un canevas Laplacien 3D+t discret pour le traitement numérique d'animations de maillages

Résumé : Dans ce rapport nous étendons le canevas Laplacien 3D discret aux animations de maillages, représentées comme des séquences temporellement cohérentes de maillages. Afin de laisser l'utilisateur contrôler l'influence du mouvement par rapport à celle de la géométrie, nous introduisons un paramètre associé à la dimension temporelle. Notre opérateur de Laplace 3D+t discret possède les mêmes propriétés que le Laplacien 3D discret, dès que ce paramètre est positif ou nul. Nous démontrons l'intérêt de ce canevas en étendant des techniques d'édition et de lissage/débruitage de maillages basées Laplacien aux animations de maillages.

Mots-clés : animation de maillages, Laplacien, édition de maillages, lissage de maillage

1 Introduction

In the last ten years, a huge amount of works have demonstrated the effectiveness of differential coordinates for mesh processing (see e.g. [Sor06] for a survey). Many of them use a discretization of the Laplace operator, which encodes the variation of a function around a given vertex. Inspired by the discrete graph Laplacian, several discrete Laplace operators have been proposed in the literature for meshed surfaces, with various properties [WMKG07, BSW08, AW11].

In this report, we propose to use the same framework for mesh animation processing. Mesh animations, also called deforming mesh sequences or 3D videos, are becoming widely used in various domains, such as entertainment, 3D television or medical simulation. As for static meshes, raw animations may require modifications, such as denoising or editing of some part of the geometry, to reach the user's objectives.

Our contribution is threefold:

- first, we decouple time and space dimensions by introducing a parameter α. Tuning this parameter allows to process either geometry or motion, or both, in a single framework;
- second, we propose a mesh animation editing method, which extends the popular static mesh editing of Sorkine et al. [SCOL*04] to time-varying meshes;
- finally, we also show how geometry and/or motion of a mesh animation can be denoised or smoothed using our discrete 3D+t Laplacian framework.

The remainder of this report is organized as follows. In section 2, we review related work on mesh and mesh animation processing. In section 3, we introduce our discrete 3D+t Laplacian framework, which is subsequently used for mesh animation editing (section 4) and fairing (section 5). We conclude in section 7.

2 Related work

2.1 Mesh animation processing

Among popular problems in mesh animation processing is mesh animation editing, for which several solutions have been proposed [KG06, SSP07, XZY*07, KG08]. In the following we use this problem as an application showcase of our discrete 3D+t Laplace operator. Tejera and Hilton's approach [TH11] is particularly close to our work, since it starts from a static Laplacian-based mesh editing framework, which is then propagated through time. Our framework is more global since we handle space and time at once. Another well studied problem is deformation transfer [SP04, BVGP09, BCWG09], in which the motion of a given animation is transferred to a new (static) mesh, in order to create a new animation. Finally, some works try to recover the underlying structure and/or motion from a given mesh sequence: for instance animation skeleton and/or skinning weights in the case of an articulated motion [JT05, DATTS08], or decomposition into pose, shape and motion [CH12]. This last approach is very interesting in the sense that it allows to process independently geometry and motion. Our purpose in this report is not to focus on a particular mesh animation processing issue, rather to propose a generic tool which may be used for many problems. It is interesting to note that many of the current approaches already use the Laplacian framework as a tool for geometry processing.

2.2 Laplacian mesh processing

Discretizations of the Laplace operator have been used for years for mesh processing. Perhaps one of the first works in this area is Taubin's fairing technique [Tau95]. Among seminal works, let us cite the implicit fairing method of Desbrun et al. [DMSB99] and the Laplacian surface editing approach of Sorkine et al. [SCOL*04]. In this report, we extend these two techniques to mesh animations. As for other interesting Laplacian mesh processing works, we refer the reader to [BKP*10, Sor06, ZvKD10].

3 Mathematical framework

3.1 The discrete 3D Laplacian framework

Laplacian mesh processing has been extensively studied for over a decade (see e.g. [Sor06, ZvKD10] for recent surveys). It relies on a discretization of the Laplace operator.

Definition 3.1 (Discrete Laplace operator [ZvKD10]) *Let $V = \{v_i\}$ be the set of vertices of a given mesh M, and $\forall i, f_i$ be the image of v_i by a function f. The* Laplace operator $\mathcal{L}$ applied to f *is such that:*

$$\forall i, (\mathcal{L}f)(v_i) = \frac{1}{d_i} \sum_{v_j \in N_i} w_{i,j}(f_i - f_j) \tag{1}$$

with N_i the 1-ring neighborhood of vertex v_i, d_i a positive factor defined for vertex v_i, and $w_{i,j}$ a weight associated to the edge $v_i v_j$.

This operator is sometimes called a *first order operator*, since we only consider 1-ring neighborhoods of vertices.

In the discrete setting, functions and operators are usually handled through vectors and matrices, respectively.

Definition 3.2 (Laplacian matrix [ZvKD10]) *The* Laplacian matrix L *for a given mesh M is $L = D^{-1}(D' - W)$, with D and D' two diagonal matrices such that:*

$$\forall i, D(i,i) = d_i,$$
$$\forall i, D'(i,i) = \sum_{v_j \in N_i} w_{i,j},$$

and W the adjacency matrix of the mesh:

$$\forall i,j, \text{ if } v_j \in N_i \text{ then } W(i,j) = w_{i,j}, \text{ else } W(i,j) = 0.$$

If $\forall i, d_i = \sum_{j \neq i} w_{i,j}$, then $D = D'$ and L can be written as $L = I - D^{-1}W$.

A particular application to vertex coordinates gives the so-called Laplacian coordinates.

Definition 3.3 (Laplacian coordinates [Sor06]) *Let v_i be a vertex of a given mesh M. The* Laplacian coordinates $\mathcal{L}(v_i)$ *of v_i are a 3-dimensional vector*

$$\mathcal{L}(v_i) = \frac{1}{\sum\limits_{v_j \in N_i} w_{i,j}} \sum_{v_j \in N_i} w_{i,j}(v_i - v_j) \tag{2}$$

with N_i the set of adjacent vertices to v_i, and $w_{i,j}$ a weight associated to the edge $v_i v_j$.

This vector can be interpreted as a discrete approximation of the surface normal at v_i.

3.2 3D+t Laplace operator

3.2.1 Notations

Let $MS = (M^1, \dots, M^m)$ be a mesh animation. Each mesh $M^k, 1 \leq k \leq m$, of the animation is a triplet (V^k, E^k, F^k), with V^k a set of vertices, E^k a set of edges, and F^k a set of faces. Actually, we only need to know the vertices of MS, that is to say the sets $V^k, \forall k \in [1, m]$, and the neighboring relationships in spacetime between them. For each vertex $v_i^k \in V^k$, we consider its coordinates in the four-dimensional spacetime: $v_i^k = (t_i^k, x_i^k, y_i^k, z_i^k)$, with t_i^k the timelike coordinate and x_i^k, y_i^k and z_i^k the spatial coordinates of vertex v_i^k. Note that for a given, static, mesh M^k, all vertices v_i^k share the same timelike coordinate: $\forall i, j, t_i^k = t_j^k = t^k$. In case the framerate f_r of the animation is constant over the sequence, $t^k = k/f_r$. Let N_i^k be the 1-ring neighborhood of vertex v_i^k. Let Ns_i^k be the subset of N_i^k made of vertices sharing the same timelike coordinate t^k as v_i^k. In other words, Ns_i^k gathers the neighbors of v_i^k in the mesh M^k. Let $Nt_i^k = N_i^k \backslash Ns_i^k$ be the "temporal" neighborhood of v_i^k.

3.2.2 Definitions

We propose the following generalization of the 3D Laplace operator to mesh animations.

Definition 3.4 (Discrete 3D+t Laplace operator) *Let $V = \{v_i^k\}$ be the set of vertices of a given mesh sequence MS, and $\forall i, k, f_i^k$ be the image of v_i^k by a function f. The* 3D+t Laplace operator $\mathcal{L}_\alpha$ applied to f *is such that:*

$$\forall i, k, (\mathcal{L}_\alpha f)(v_i^k) = \frac{\alpha}{\delta_i^k} \sum_{v_j^l \in Nt_i^k} w_{i,j}^{k,l}(f_i^k - f_j^l) + \frac{1}{d_i^k} \sum_{v_j^k \in Ns_i^k} w_{i,j}^k (f_i^k - f_j^k) \tag{3}$$

with δ_i^k and d_i^k two positive factors defined for vertex v_i^k, $w_{i,j}^{k,l}$ a weight associated to the edge $v_i^k v_j^l$, and $w_{i,j}^k$ a weight associated to the edge $v_i^k v_j^k$. α is a user-defined parameter (see Section 3.2.4).

Note that this definition can be thought of as a discretization of the d'Alembert operator, which is the counterpart of the Laplace operator in the Minkowski spacetime [Che04]. However, usual metrics in the Minkowski spacetime, with an opposite sign between time and space dimensions, lead to a hyperbolic operator, while our 3D+t Laplace operator remains an elliptic one as long as the scale factor α remains non negative.

The previous definition 3.4 is very general and can be applied to any mesh sequence. However, in the rest of this report, we restrict to mesh sequences which are explicitly temporally coherent, that is to say with a fixed connectivity: $\forall k_1 \neq k_2, v_j^{k_1} \in Ns_i^{k_1} \iff v_j^{k_2} \in Ns_i^{k_2}$. Most deforming mesh sequences used in computer graphics are temporally coherent, since they are constructed from a single mesh which deforms over time. Temporally coherent mesh sequences can also be generated from multi-view video systems [DAST*08, VBMP08].

In the case of temporally coherent mesh sequences, only two "temporal" weights $w_{i,j}^{k,l}$ are defined for each vertex v_i^k: $(j,l) = (i, k-1)$ or $(i, k+1)$. Let us denote these weights w_i^{k-} and w_i^{k+}. The discrete 3D+t Laplace operator can now be written the following way.

Definition 3.5 (Discrete 3D+t Laplace operator for temporally coherent mesh sequences) *Let $V = \{v_i^k\}$ be the set of vertices of a given temporally coherent mesh sequence MS, and $\forall i, k, f_i^k$ be the image of v_i^k by a function f. The* 3D+t Laplace operator $\mathcal{L}_\alpha$ applied to f *is such that:*

$$\forall i, k, (\mathcal{L}_\alpha f)(v_i^k) = \frac{\alpha}{\delta_i^k}(w_i^{k-}(f_i^k - f_i^{k-1}) + w_i^{k+}(f_i^k - f_i^{k+1}))$$

$$+\frac{1}{d_i^k}\sum_{v_j^k \in Ns_i^k} w_{i,j}^k (f_i^k - f_j^k) \tag{4}$$

with d_i^k and δ_i^k two positive factors defined for vertex v_i^k, $w_{i,j}^k$ a weight associated to the edge $v_i^k v_j^k$ and w_i^{k-} and w_i^{k+} weights associated to the temporal edges $v_i^k v_i^{k-1}$ and $v_i^k v_i^{k+1}$, respectively.

Figure 1 illustrates the difference between the two definitions. In the second case, there is a one-to-one correspondence between vertices of two successives meshes M^k and M^{k+1}. Note that our discretization is a first order one, since our temporal weights are only defined for the two contiguous frames $k-1$ and $k+1$ of the current frame k.

This discrete 3D+t Laplace operator for temporally coherent mesh sequences can also be handled matricially. Let us note n the number of vertices of each mesh M^k of a temporally coherent sequence MS. The total number of vertices in MS is then nm. We therefore handle much bigger matrices than in the Laplacian processing case ($nm \times nm$ instead of $n \times n$), but fortunately these matrices are very sparse.

Definition 3.6 (3D+t Laplacian matrix) *The* 3D+t Laplacian matrix L_α *for a given temporally coherent mesh sequence MS is the $nm \times nm$ matrix*

$$L_\alpha = \alpha \Delta^{-1}(\Delta' - W_t) + D^{-1}(D' - W_s), \tag{5}$$

with:

- *Δ a $nm \times nm$ diagonal matrix such that $\forall k \in [1, m], \forall i \in [1, n], \Delta((k-1)n + i, (k-1)n + i) = \delta_i^k$,*
- *Δ' a $nm \times nm$ diagonal matrix such that $\forall k \in [1, m], \forall i \in [1, n], \Delta'((k-1)n + i, (k-1)n + i) = w_i^{k-} + w_i^{k+}$,*

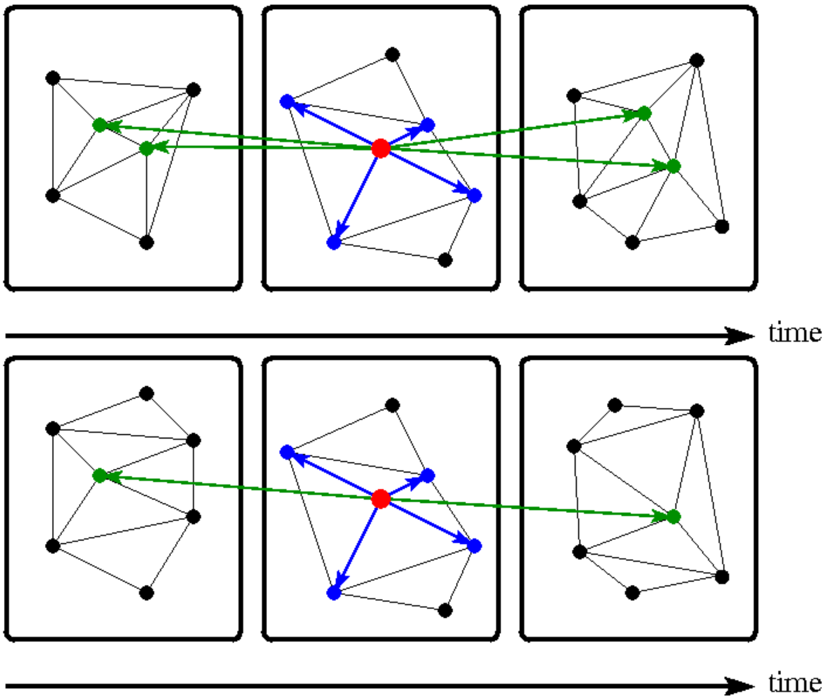

Figure 1: Neighboring vertices involved in the definition of the discrete 3D+t Laplace operator for a given vertex v_i^k. Top: general definition (Equation (3)). Bottom: temporally coherent mesh sequence case (Equation (4)). In both cases the red dot corresponds to v_i^k, the blue dots to spatial neighbors v_j^k and the green dots to temporal neighbors v_j^{k-1} and v_j^{k+1}.

- W_t *a* $nm \times nm$ *sparse matrix such that* $\forall k \in [2, m], \forall i \in [1, n], W_t((k-1)n + i, (k-2)n + i) = w_i^{k-}$ *and* $\forall k \in [1, m-1], \forall i \in [1, n], W_t((k-1)n + i, kn + i) = w_i^{k+}$, *otherwise* $\forall i, j \in [1, nm], W_t(i, j) = 0$,
- D *a* $nm \times nm$ *diagonal matrix such that* $\forall k \in [1, m], \forall i \in [1, n], D((k-1)n + i, (k-1)n + i) = d_i^k$,
- D' *a* $nm \times nm$ *diagonal matrix such that* $\forall k \in [1, m], \forall i \in [1, n], D'((k-1)n + i, (k-1)n + i) = \sum_{v_j^k \in Ns_i^k} w_{i,j}^k$,
- W_s *a* $nm \times nm$ *block-diagonal matrix made of* m $n \times n$ *sub-matrices* W^k, *which are the spatial adjacency matrices of meshes* M^k.

Notice that $D^{-1}(D' - W_s)$ is a block-diagonal matrix made of the (spatial) Laplacian matrices L^k of meshes M^k. Let J^k be the diagonal $n \times n$ matrix such that $\forall i, J^k(i, i) = \frac{w_i^{k-} + w_i^{k+}}{\delta_i^k}$. If we denote W^{k-} the diagonal $n \times n$ matrix such that $\forall i, W^{k-}(i, i) = \frac{w_i^{k-}}{\delta_i^k}$ and W^{k+} the diagonal $n \times n$ matrix such that $\forall i, W^{k+}(i, i) = \frac{w_i^{k+}}{\delta_i^k}$, then L_α can be written blockwise as:

$$\begin{bmatrix} \alpha J^1 + L^1 & -\alpha W^{1+} & & & \\ -\alpha W^{2-} & \alpha J^2 + L^2 & -\alpha W^{2+} & & \\ & \ddots & \ddots & \ddots & \\ & & -\alpha W^{(m-1)-} & \alpha J^{m-1} + L^{m-1} & -\alpha W^{(m-1)+} \\ & & & -\alpha W^{m-} & \alpha J^m + L^m \end{bmatrix}$$

Since L_α is very sparse, it can usually be inverted very efficiently, as demonstrated in Sections 4 and 5.

3.2.3 3D+t Laplacian coordinates

Laplacian coordinates can also be generalized to mesh sequences, as a particular application of the previously defined discrete 3D+t Laplace operator to vertex coordinates in spacetime.

Definition 3.7 (3D+t Laplacian coordinates) *Let* v_i^k *be a vertex of a given temporally coherent mesh sequence* MS. *The* 3D+t Laplacian coordinates $\mathcal{L}_\alpha(v_i^k)$ *of* v_i^k *are a 4-dimensional vector*

$$\mathcal{L}_\alpha(v_i^k) = \frac{\alpha}{w_i^{k-} + w_i^{k+}}(w_i^{k-}(v_i^k - v_i^{k-1}) + w_i^{k+}(v_i^k - v_i^{k+1})) + \frac{1}{\sum_{v_j^k \in Ns_i^k} w_{i,j}^k} \sum_{v_j^k \in Ns_i^k} w_{i,j}^k (v_i^k - v_j^k)) \tag{6}$$

with Ns_i^k *the set of adjacent vertices to* v_i^k *in space,* $w_{i,j}^k$ *a weight associated to the edge* $v_i^k v_j^k$, *and* w_i^{k-} *and* w_i^{k+} *weights associated to the temporal edges* $v_i^k v_i^{k-1}$ *and* $v_i^k v_i^{k+1}$, *respectively.*

As the Laplacian coordinates in $\mathbb{R}^3$, they can be interpreted in spacetime as a discrete approximation of the mesh sequence normal at v_i^k.

3.2.4 Weights and parameter

As for the discrete 3D Laplace operator, different weight choices lead to different properties [WMKG07, BSW08]. Spatial weights $w_{i,j}^k$ can be either purely combinatorial, or geometrical, such as the famous cotangent weights [DMSB99]. A basic solution to set temporal weights is to use the 3-point stencil finite difference method, which leads to $w_i^{k-} = w_i^{k+} = 1$. However, this is only meaningful for temporally coherent mesh sequences with constant framerate. Otherwise, temporal weights should be set to $w_i^{k-} = c$ and $w_i^{k+} = \frac{t^k - t^{k-1}}{t^{k+1} - t^k} c$, with c a user-chosen constant. In most applications, d_i^k is set to $d_i^k = \sum_{v_j^k \in Ns_i^k} w_{i,j}^k$ and δ_i^k is set to $\delta_i^k = w_i^{k-} + w_i^{k+}$.

The α scale factor balances the influence of temporal and spatial neighbors over a vertex. When α goes to zero, our discrete 3D+t Laplace operator acts exactly as a spatial Laplace operator. Temporal neighbors have no influence on a given vertex. On the contrary, when α goes to infinity, the 3D+t Laplace operator acts as a temporal averaging operator, and spatial neighbors have no influence on a given vertex. Aganj et al., by considering a scaling factor between time and space dimensions, proposed a similar solution [APSK07].

Our discrete 3D+t Laplace operator can easily be used for various mesh animation processing tasks. In the next two sections, we show how two famous mesh processing techniques (mesh editing by Sorkine et al. [SCOL*04] and mesh fairing by Desbrun et al. [DMSB99]) can be extended to mesh animations, simply by replacing the 3D Laplace operator with the 3D+t Laplace one. By tuning the α parameter, we can then process either the shape or the motion of an existing mesh animation. Note that for both applications vertex timelike coordinates are naturally smoothed. To counterbalance this, we simply add temporal constraints to the linear system we solve: $\forall k \neq K, \forall i, \|{t'}_i^k - t^k\|^2 = 0$.

4 Application to mesh animation editing

We now describe a solution to edit a mesh sequence using the previously defined discrete 3D+t Laplace operator. We restrict to purely geometric mesh animation editing. This means that, for every vertex of the mesh sequence, its timelike coordinate should not be modified. This is indeed what is usually required by infographists, since modifying the vertex timelike coordinates may break the static mesh connectivities.

4.1 Method

In this work, we take inspiration from [SCOL*04]. We fix the absolute positions u_i of several vertices $v_i^K, p \leq i \leq n$, all belonging to the same mesh M^K, and we solve for the remaining vertices of the sequence. More precisely, we minimize the following error functional:

$$E(V') = \sum_{k=1}^{m} \sum_{i=1}^{n} \|T_i^k(V')\mathcal{L}_\alpha(v_i^k) - \mathcal{L}_\alpha({v'}_i^k)\|^2 + \sum_{i=p}^{n} \|{v'}_i^K - u_i\|^2 \tag{7}$$

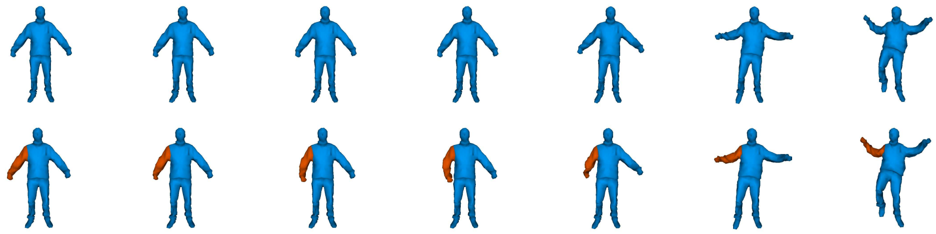

Figure 2: Editing the *Crane* animation [VBMP08]. First row: input sequence. Second row: edited sequence. Vertices in the spatial region of interest appears in orange. The temporal window is 21-frame long. Successive images correspond to frames 1, 5, 8, 11, 14, 17 and 21 respectively.

with $\forall i, k, v_i^k$ representing the initial position of the vertices and ${v'}_i^k$ the (unknown) final positions of theses vertices. V' is a $nm \times 4$ matrix containing the ${v'}_i^k$, and T_i^k is an (unknown) transformation for vertex v_i^k which minimizes

$$\|T_i^k v_i^k - {v'}_i^k\|^2 + \sum_{v_j^l \in N_i^k} \|T_i^k v_j^l - {v'}_j^l\|^2$$

Similarly to [SCOL*04], if we want to include geometric translations, rotations and scaling and keep all timelike coordinates constant, any T_i^k can be expressed in homogeneous coordinates as a 5×5 matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & s & -h_3 & h_2 & t_x \\ 0 & h_3 & s & -h_1 & t_y \\ 0 & -h_2 & h_1 & s & t_z \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

This leads to the exact same minimization $\|A_i^k (s_i^k, h_i^k, t_i)^T - b_i^k\|^2$ as in [SCOL*04].

We implemented this extension of [SCOL*04], minimizing the energy defined by Equation (7). The user selects a geometrical region of interest (ROI) in mesh M^K, but also a temporal window $[K - dt, K + dt]$ around frame nr. K: meshes M^1 to M^{K-dt-1} and M^{K+dt+1} to M^m are not modified. Similarly to [SCOL*04], we set as additional soft constraints geometrical stationary anchors (the boundary of the ROI, for all meshes in the temporal window) but also temporal stationary anchors (the entire ROI, for the boundary meshes M^{K-dt} and M^{K+dt}).

4.2 Results

Figure 2 shows a result of this editing method, where we move left the right arm of the character at frame 11. Other frames are consequently modified. For all results shown in this section, we used the cotangent weights as spatial weights $w_{i,j}^k$, and 1 as temporal weights w_i^{k-} and w_i^{k+}, since the framerate of the input animation is constant.

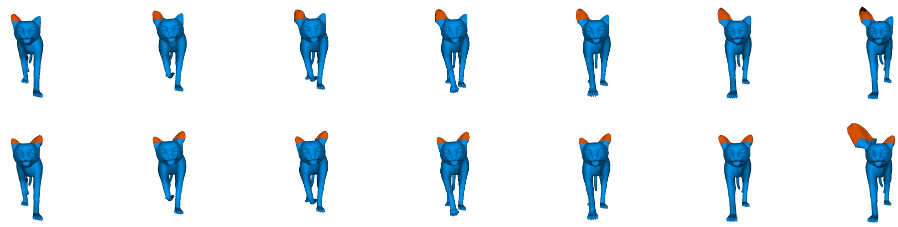

Figure 3: Creating a bunny ear on a walking cat. First row: $\alpha = 1$ for the right ear; left ear is not modified. Second row: $\alpha = 100$ for the right ear, $\alpha = 0.01$ for the left ear. Modifications are applied on a 33-frame long temporal window. Successive images correspond to frames 1, 7, 9, 11, 13, 15 and 17 respectively.

4.2.1 Tuning the parameter

The α parameter can be tuned to obtain various editing effects, see Figure 3. When α is low (but remains non negative), the temporal neighborhood does not affect much the change of vertex coordinates, and the 3D+t Laplace operator acts as a classical 3D Laplace operator. Thus, mesh M^K at frame K is modified but few other meshes in the temporal window $[K - dt, K + dt]$ are modified. On the contrary, when α is high, the 3D+t Laplace operator acts more as a temporal smoothing operator. Spatial coordinates of any vertex v_i^K are modified according to the spatial coordinates of the same vertex for neighboring frames. If these coordinates are similar, then the vertex position remains approximately constant. Temporal weights may also be tuned. To get a symmetrical influence from past and future frames, any pair of weights (w_i^{k-}, w_i^{k+}) should be set with respect to the animation's current framerate: $w_i^{k-}(t^k - t^{k-1}) + w_i^{k+}(t^k - t^{k+1}) = 0$. In all of our experiments, the framerate is constant over the whole sequence. We thus have $\forall k, t^{k+1} - t^k = t^k - t^{k-1}$, and we set $\forall i, \forall k, w_i^{k-} = w_i^{k+} = 1$. We could set the w_i^{k-} to another value, but since they are directly related to α, we find it more convenient to tune α directly.

4.2.2 The wave effect

Using a negative value for the α parameter leads to a negative influence of temporal neighbors on a given vertex. Induced deformation goes back and forth, as shown on Figure 4. This phenomenon, which we call *the wave effect*, can be naturally explained since our Equation (7) is related to the so-called *wave equation* $\Box f = 0$, whose solutions are waves propagating through the ambient spacetime. Actually, minimizing Equation (7) is quite similar to solve for a discretization of the wave equation, with boundary conditions.

Note that this application is only intended to demonstrate the usefulness of the discrete 3D+t Laplace operator. As a basic extension of Sorkine et al.'s method, it suffers from the same drawbacks. In particular, it is only valid for rotations with small angles. Better results may be obtained using other static deformation methods, see [BS08] for a survey.

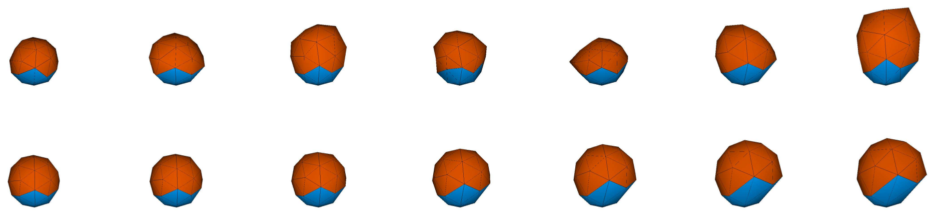

Figure 4: The wave effect. First row: deforming an initially static sphere (i.e., a sequence where the same mesh is duplicated) to reach the top right corner with $\alpha < 0$ (here, $\alpha = -1$) generates vibrations. Second row: using a positive value for α (here, $\alpha = 1$) leads to a smooth deformation. The temporal window is 91-frame long; successive images correspond to frames 28, 31, 34, 37, 40, 43 and 46 respectively.

5 Application to mesh animation fairing

Our next application of the discrete 3D+t Laplace operator is for mesh animation fairing (smoothing or denoising). In this work, we extend the framework developed by [DMSB99] to mesh animations, in order to fair either its shape or its motion.

5.1 Method

The basic idea of Laplacian fairing of a static mesh is to solve Laplace's equation $\triangle f = 0$ for coordinates. This is done incrementally by integrating the heat equation $\frac{\partial f}{\partial t} - \lambda \triangle f = 0$: if $V^0 = V$ is the vector made of the mesh's vertices, then at each time step t a new vector V^t is derived such that $V^t = (I + \lambda dtL)V^{t-1}$, with dt a parameter controlling the diffusion speed [DMSB99].

This operation can be generalized to mesh sequences. Let V^t be a vector containing the nm vertices of the sequence (ordered by their timelike coordinates: first, the vertices of the first mesh, then the vertices of the second mesh, etc.) after t fairing steps. The new diffusion process can simply be written as

$$V^t = (I + \lambda dtL_\alpha)V^{t-1}, \tag{8}$$

with L_α the 3D+t Laplacian matrix as described in Definition 3.6. This process modifies not only the spatial, but also the timelike coordinates of the vertices of each mesh. According to Equation (4) applied to the timelike coordinates, a solution to enforce the timelike coordinates to remain constant is to set:

$$\forall i, w_i^{k-}(t^k - t^{k-1}) + w_i^{k+}(t^k - t^{k+1}) = 0$$

Any choice of w_i^{k-} and w_i^{k+} such that $w_i^{k+} = \beta w_i^{k-}$, with $\beta = \frac{t^k - t^{k-1}}{t^{k+1} - t^k}$, is a sufficient condition for this equation to hold.

As in [DMSB99], we choose to rather solve the implicit equation $(I + \lambda dtL_\alpha^2)V^t = V^{t-1}$ in order to get rid of integration time step limitations. We also add temporal constraints to the system: $\forall k, \forall i, \|{t'}_i^k - t_i^k\|^2 = 0$, in order each timelike coordinate to remain constant. We use the cotangent weights as spatial weights $w_{i,j}^k$. Note that other Laplacian-based mesh smoothing schemes, such as [NSACO06], may be adapted for mesh sequences using the same framework.

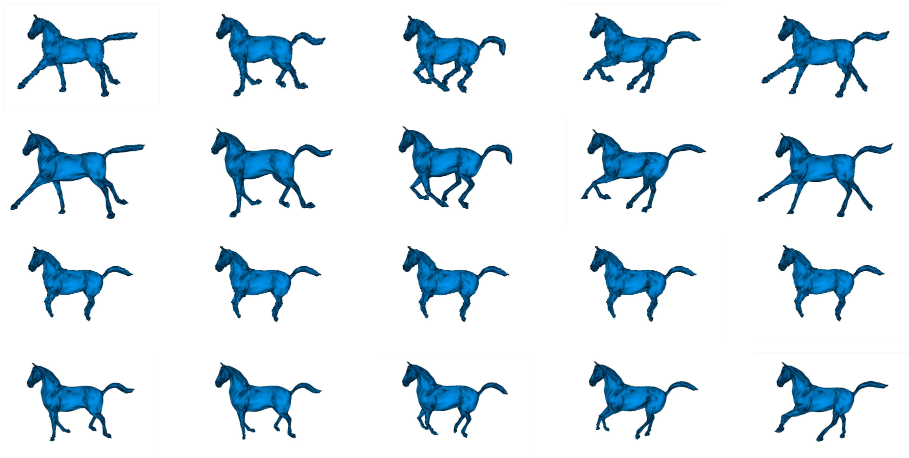

Figure 5: Fairing the geometry and/or the motion of a horse. First row: input sequence. Second row: $(\alpha, \lambda dt) = (0.01, 10)$. Third row: $(\alpha, \lambda dt) = (100, 0.1)$. Fourth row: $(\alpha, \lambda dt) = (1, 10)$. For each row, successive images correspond to frames 1, 3, 6, 9 and 12 respectively.

5.2 Results

Figure 5 shows fairing results on the *Horse gallop* sequence [SP04], to which random geometric noise has been added. Since the framerate is constant along the whole sequence, we set $w_i^{k+} = w_i^{k-} = 1$ for all k. 5 iterations were performed in each case. Each iteration took about 5 seconds (in the first and third case) or 24 seconds (in the second case) to solve, for each spatial coordinate. Tuning the α parameter, we can either denoise the geometry, smooth the motion (up to freeze it to a static mean pose), or do a mix of both. Figure 6 shows another fairing result, for which only 1 iteration was performed.

6 Performances

The two presented applications have been implemented in C++, using the Eigen library (*http://eigen.tuxfamily.org/*). We used the UmfPack version of the LU matrix decomposition for the edit application, and the biconjugate gradient solver for the fairing application.Timings are given below for a standard low-end laptop with 1.6 GHz processor. In the following table, m is the number of meshes in the sequence, and n is the number of vertices for each mesh of the sequence. The system matrix is usually big but very sparse, which leads to reasonable computation times.

Sequence	m	n	Matrix size	Non zeros	Overall
Fig. 2	21	143	12012^2	0.68%	11s
Fig. 4	91	29	10556^2	0.62%	1s
Fig. 3	33	90	10556^2	0.68%	5s
Fig. 5 (1, 3)	12	8431	101172^2	0.21%	96s
Fig. 5 (2)	12	8431	101172^2	0.21%	381s
Fig. 6	175	1001	175175^2	1.02%	157s

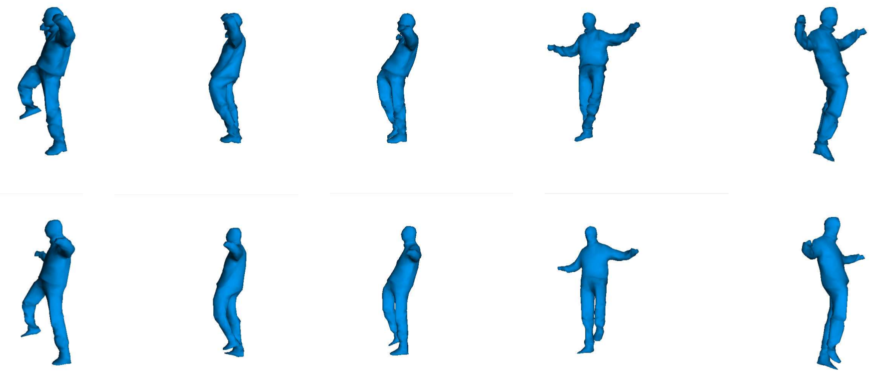

Figure 6: Fairing both geometry and motion of the Crane animation [VBMP08]. First row: input sequence. Second row: $(\alpha, \lambda dt) = (10, 10)$. For each row, successive images correspond to frames 24, 48, 72, 96 and 120 respectively.

7 Conclusion

We have proposed in this report a unified framework for mesh animation geometry and motion processing. We have defined a discretization of a 3D+t Laplace operator. As for other discrete Laplace operators, this operator is easily expressed by a sparse matrix (rarely containing more than 1% non zero coefficients, according to our experiments). This yields to simple and efficient computations. Introducing a user-defined parameter α allows to control the motion influence with respect to the geometry. As a first step towards 3D+t Laplace-based mesh animation processing, we extended seminal Laplace-based mesh editing and smoothing techniques to mesh animations.

We hope our discrete 3D+t Laplace operator will inspire other interesting mesh animation processing works. Our main concern for future work is about the spectral analysis of this operator. Indeed, Laplacian eigenvalues and eigenvectors have been successfully considered for shape matching, classification, pose-invariant segmentation, compression, etc., due to their interesting properties [ZvKD10]. We are interested in investigating similar properties for the 3D+t Laplace eigenspectrum, whatever the value of the parameter α is.

Other perspectives include an automatic tuning of the α parameter, depending on the application.

Acknowledgements

The author thanks Rémy Cumont and Léo Blin (Grenoble INP - Ensimag students) for a preliminary version of the mesh animation editing code, Benjamin Aupetit for advices using Eigen, Edmond Boyer and Lionel Revéret for proofreading. This work is partially supported by the ANR through the MORPHO project (ANR-10-BLAN-0206).

References

[APSK07] AGANJ E., PONS J.-P., SÉGONNE F., KERIVEN R.: Spatio-temporal shape from silhouette using four-dimensional delaunay meshing. In *IEEE International Conference on Computer Vision* (2007).

[AW11] ALEXA M., WARDETZKY M.: Discrete laplacian on general polygonal meshes. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 30*, 4 (2011).

[BCWG09] BEN-CHEN M., WEBER O., GOTSMAN C.: Spatial deformation transfer. In *Symposium on Computer Animation* (2009).

[BKP*10] BOTSCH M., KOBBELT L., PAULY M., ALLIEZ P., LÉVY B.: *Polygon mesh processing*. A.K. Peters, 2010.

[BS08] BOTSCH M., SORKINE O.: On linear variational surface deformation methods. *IEEE Trans. on Vis. and Computer Graphics 14*, 1 (2008).

[BSW08] BELKIN M., SUN J., WANG Y.: Discrete laplace operator on meshed surfaces. In *Symposium on Computational Geometry* (2008).

[BVGP09] BARAN I., VLASIC D., GRINSPUN E., POPOVIĆ J.: Semantic deformation transfer. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 28*, 4 (2009).

[CH12] CASHMAN T., HORMANN K.: A continuous, editable representation for deforming mesh sequences with separate signals for time, pose and shape. *Computer Graphics Forum (Proc. of Eurographics) 31*, 2 (2012).

[Che04] CHENG T.-P.: *Relativity, gravitation and cosmology: a basic introduction*. Oxford University Press, 2004.

[DAST*08] DE-AGUIAR E., STOLL C., THEOBALT C., AHMED N., SEIDEL H.-P., THRUN S.: Performance capture from sparse multi-view video. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 27*, 3 (2008).

[DATTS08] DE-AGUIAR E., THEOBALT C., THRUN S., SEIDEL H.-P.: Automatic conversion of mesh animations into skeleton-based animations. *Computer Graphics Forum (Proc. of Eurographics) 27*, 2 (2008).

[DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. *ACM Computer Graphics (Proc. of SIGGRAPH)* (1999).

[JT05] JAMES D., TWIGG C. D.: Skinning mesh animations. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 24*, 3 (2005).

[KG06] Kircher S., Garland M.: Editing arbitrarily deforming surface animations. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 25*, 3 (2006).

[KG08] Kircher S., Garland M.: Free-form motion processing. *ACM Trans. on Graphics 27*, 2 (2008).

[NSACO06] Nealen A., Sorkine O., Alexa M., Cohen-Or D.: Laplacian mesh optimization. In *ACM GRAPHITE* (2006).

[SCOL*04] Sorkine O., Cohen-Or D., Lipman Y., Alexa M., Rössl C., Seidel H.-P.: Laplacian surface editing. In *Symposium on Geometry Processing* (2004).

[Sor06] Sorkine O.: Differential representations for mesh processing. *Computer Graphics Forum 25*, 4 (2006), 789–807.

[SP04] Sumner R. W., Popović J.: Deformation transfer for triangle meshes. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 23*, 3 (2004).

[SSP07] Sumner R. W., Schmid J., Pauly M.: Embedded deformation for shape manipulation. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 26*, 3 (2007).

[Tau95] Taubin G.: A signal processing approach to fair surface design. *ACM Computer Graphics (Proc. of SIGGRAPH)* (1995).

[TH11] Tejera M., Hilton A.: Space-time editing of 3d video sequences. In *Proc. of the European Conference on Visual Media Production (CVMP)* (2011).

[VBMP08] Vlasic D., Baran I., Matusik W., Popović J.: Articulated mesh animation from multi-view silhouettes. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 27*, 3 (2008).

[WMKG07] Wardetzky M., Mathur S., Kälberer F., Grinspun E.: Discrete laplace operators: no free lunch. In *Symposium on Geometry Processing* (2007).

[XZY*07] Xu W., Zhou K., Yu Y., Tan Q., Peng Q., Guo B.: Gradient domain editing of deforming mesh sequences. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 26*, 3 (2007).

[ZvKD10] Zhang H., van Kaick O., Dyer R.: Spectral mesh processing. *Computer Graphics Forum 29*, 6 (2010).

Contents

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399